OIPE IAP84
JUL 05 2006
PATENT & TRADEMARK OFFICE

07-07-06

IEW 2137

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re the Application of:

REISMAN, Arthur

Serial No.: 09/435,736

Filed: November 8, 1999

Atty. File No.: 4366-41

For: "ENCRYPTED AND NON-ENCRYPTED COMMUNICATION OF MESSAGE DATA"

Group Art Unit: 2137

Examiner: NGUYEN, M.

SUBMISSION OF DECLARATION OF DAVID VOLEJNICEK UNDER 37 C.F.R. §1.131

"EXPRESS MAIL" MAILING LABEL NUMBER: EV 788583696 US
DATE OF DEPOSIT: July 5, 2006

I HEREBY CERTIFY THAT THIS PAPER OR FEE IS BEING DEPOSITED WITH THE UNITED STATES POSTAL SERVICE "EXPRESS MAIL POST OFFICE TO ADDRESSEE" SERVICE UNDER 37 CFR 1.10 ON THE DATE INDICATED ABOVE AND IS ADDRESSED TO THE COMMISSIONER FOR PATENTS, P.O. BOX 1450, ALEXANDRIA, VA 22313-1450

TYPED OR PRINTED NAME: Christine Jacquet

SIGNATURE: Christine Jacquet

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Enclosed is a Declaration of David Volejnicek Under 37 C.F.R. §1.131 (including Exhibits A-E) , filed to support the Amendment and Response filed on June 30, 2006, in connection with the above-identified application.

Respectfully submitted,

SHERIDAN ROSS P.C.

By: Douglas Swartz

Douglas W. Swartz
Registration No. 37,739
1560 Broadway, Suite 1200
Denver, Colorado 80202-5141
(303) 863-9700

Date: July 5, 2006

OIPE IAP84
JUL 05 2006
PATENT & TRADEMARK OFFICE

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re the Application of:	)	Group Art Unit: 2137
REISMAN, ARTHUR	)	Examiner: NGUYEN, M.
Serial No.: 09/435,736	)	DECLARATION OF DAVID VOLEJNICEK UNDER 37 C.F.R. §1.131
Filed: November 8, 1999	)	
Atty. File No.: 4366-41	)	
For: "ENCRYPTED AND NON-ENCRYPTED COMMUNICATION OF MESSAGE DATA"	)	

"EXPRESS MAIL" MAILING LABEL NO: EV 788583696 US
DATE OF DEPOSIT: July 5, 2006

I HEREBY CERTIFY THAT THIS CORRESPONDENCE IS BEING DEPOSITED WITH THE UNITED STATES POSTAL SERVICE "EXPRESS MAIL POST OFFICE TO ADDRESSEE" SERVICE UNDER 37 C.F.R. 1.10 ON THE DATE INDICATED ABOVE AND IS ADDRESSED TO THE COMMISSIONER FOR PATENTS, P.O. BOX 1450, ALEXANDRIA, VA 22313-1450.

TYPED OR PRINTED NAME: Christine Jacquet

SIGNATURE: Christine Jacquet

Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

Dear Sir:

I, David Volejnicek, declare as follows:

1. I am corporate counsel of Avaya Technology Corp., the current assignee of the above application, and a former corporate counsel of Lucent Technologies, Inc. ("Lucent"), the former assignee of the above application. I am a shareholder in Lucent and in Avaya, Inc., the parent of Avaya Technology Corp. This Declaration is being submitted in connection with prosecution activities for the above-referenced patent application.

2. Upon information and belief, Patent Submission 116621, attached hereto as Exhibit "A" and entitled "Sparse Web Encryption with Lowest Common Denominator Java" was received for consideration by the Intellectual Property Law department of Lucent by October 8, 1998, as evidenced by the letter to the inventor attached hereto as Exhibit "D". Upon information and belief, the IP Law department approved the submission for filing as a US patent application by February 22, 1999, as evidenced by the IP Law department form attached hereto as Exhibit "E".

3. Upon information and belief, Eric Grosse, an employee of Lucent, sent an email dated September 29, 1997, to the inventor opining on the invention. Upon information and belief, by emails dated September 30, 1997, the inventor sent Jack Penrod, corporate counsel of Lucent, a description of the invention. Upon information and belief, various further emails were exchanged between Penrod and the inventor on September 30, 1997. These emails are attached hereto as Exhibit "B". Upon information and belief, a copy of the submission was annotated by hand on October 7, 1998. A copy of the annotated submission is attached hereto as Exhibit "C".

4. Upon information and belief, by letter dated June 7, 1999, the patent submission was forwarded by Eli Weiss, corporate counsel of Lucent, to outside counsel. The requested filing deadline was September 15, 1999. Upon information and belief, by email dated July 19, 1999, outside counsel forwarded to the inventor materials to assist the inventor in providing input to outside counsel before the patent application was prepared. Upon information and belief, Penrod emailed outside counsel on July 22, 1999, indicating that he had hoped to prepare a patent application directed to the submission himself but was unable to due to time constraints. Upon information and belief, during the period from September 13, 1999, to October 4, 1999, outside counsel and the inventor exchanged numerous voice messages in an attempt to discuss technical aspects of the invention. Upon information and belief, an interview between outside patent counsel and the inventor was held by October 5, 1999. Upon information and belief, on October 11, 1999, outside counsel forwarded, by email, proposed claims to the inventor for review. Upon information and belief, a revised set of claims was emailed to the inventor on October 14, 1999. Upon information and belief, a draft of the proposed application and figures were sent to the inventor by October 19, 1999. Upon information and belief, the inventor approved the application by October 20, 1999. Upon information and belief, outside counsel forwarded on October 19, 1999, an Attachment H to Lucent requesting a case name and number. Upon information and belief, the finalized application and

figures, and assignment and Declaration/Power of Attorney for execution, were sent to the inventor on November 3, 1999. The subject application was thereafter filed on November 8, 1999.

4. With reference to independent claims 36, 40, 44, and 45, Exhibits "A" and "B" describe an HTTP browser and server communicating over the Internet. The browser includes an HTML form and a JAVA application to encrypt sensitive fields using the public key. A TCP/IP socket carries the encrypted data. The server, in the unique client session, generates the HTML and provides the JAVA application (which is downloaded to the browser). The server further has a set of private and public keys that are used to decrypt the encrypted fields. The encryption is sparse in that only those pieces of information in the HTML form requiring encryption (e.g., passwords, credit card numbers, phone numbers, etc.) are encrypted. In contrast, SSL servers encrypt everything, requiring much more horse power on the server and client sides. Client encryption uses the most basic JAVA utilities and a very simple JAVA application. This can eliminate common client/browser differences with full blown JAVA applications. A dynamic two key public/private encryption ensures that only the financial/merchant institution has the decryption key.

5. With further reference to Exhibits "A" and "B", a basic scenario is described in which: (1) the customer logs into a web site to purchase an item; (2) the customer decides what he wants and follows the link to an HTML order form page; (3) on the purchase form embedded in the HTML page is a JAVA applet that presents the sensitive data field(s) to the customer (the bulk of the page, and all its graphics, are standard HTML); (4) the JAVA applet contains the encryption algorithm and unique encryption public key created dynamically at the server each time the page is accessed (the server also dynamically creates a matching private key, with the encryption algorithm being chosen from the domain of available private/public key schemes); (5) the JAVA applet opens a socket back to the merchant's server and sends the encrypted sensitive information back to the server; (6) the server holds the unique matching private decryption key for the transaction (it also

passed a session ID (cookie) to the client so that it matches the correct decryption key); and (7) the server securely decrypts the sensitive information and fills the customer's order.

6. The foregoing statements and attached exhibits establish a conception date before the September 1, 1999, filing date of U.S. 6,772,333 to Brendel and the March 18, 1999, priority date of U.S. 6,529,885 to Johnson, and diligence between the conception date and the constructive reduction-to-practice date, or the filing date of the above-captioned application.

7. I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further, that the statements were made with the knowledge that willful false statements and the like, so made, are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the subject application or any patent issuing thereon.

Date: 5 JULY 2006

By: David Volejnicek

David Volejnicek
Reg. No. 29,355

EXHIBIT A

IHPAT 98-113

Sparse Web Encryption with lowest common denominator Java

Art Reisman
Lucent Technologies

Sparse-

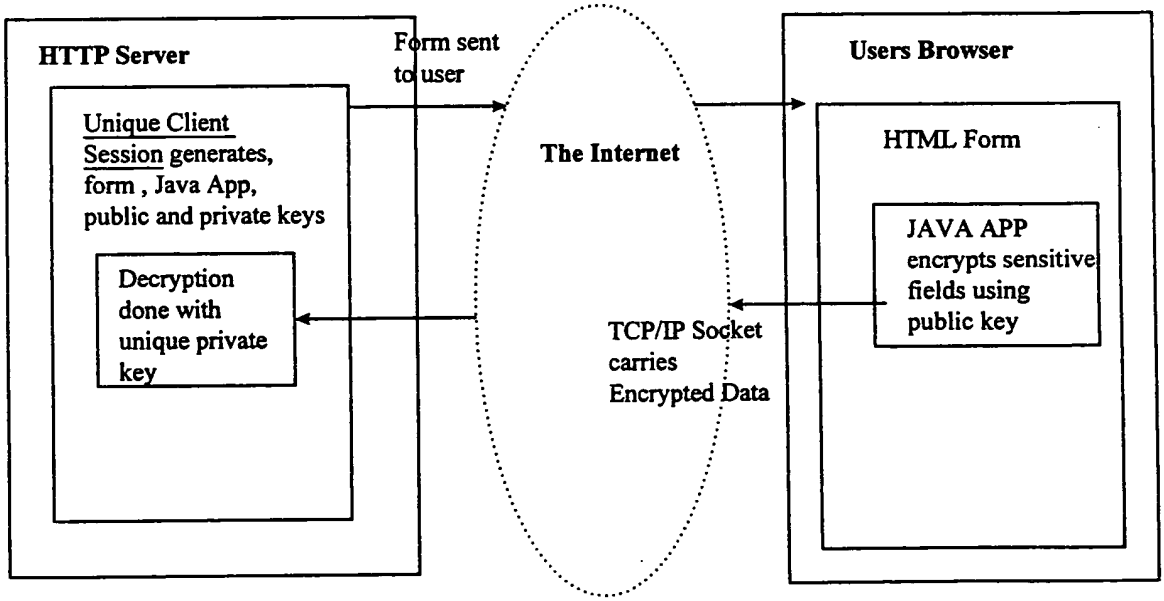
You only encrypt those pieces of information that require it (Passwords, Credit Card numbers, Phone Numbers. SSL servers generally encrypt everything requiring much more horse power on the server and client side.

Lowest Common Denominator Java-

Client Encryption Uses the most basic Java Utilities and a very simple Java application. This eliminates common client browser differences with full blown Java Applications. The delays experienced with larger Java Applications are not encountered

Dynamic two key public/private encryption insures that only the financial/merchant institution has the decryption key.

Sparse Encryption Scenario



Basic scenario

1. Customer logs into web site to purchase an item.

2. The customer decides what they want , and follows the link to an HTML order form page.

3. On the purchase form embedded in the HTML page is a Java Applet that presents the sensitive data field(s) to the customer. The bulk of the page, and all its graphics are standard HTML, (hence the term sparse encryption)

4. The Java applet contains the encryption algorithm and unique encryption public key created dynamically at the server each time the page is accessed. The server also dynamically creates a matching private key. The encryption algorithm will be chosen from the domain of available private/public key schemes and may not be unique to this implementation[1]

5. The JAVA applet opens a socket back to merchants server and sends the encrypted sensitive information[2] back to the server.

6. The server holds the unique matching private decryption key for this transaction. It also passed a session id (cookie) to the client so it match the correct decryption key.

7. The server securely decrypts the sensitive information and fills the customer order.

FAQ

Q : What if a random surfing internet hacker picks off the order form with the encryption key. Can the hacker pretend to be the customer.

A: In this scenario the hacker would be able to see everything except the sensitive information. The hacker cannot see the sensitive information entered by the actual customer as it is only decrypted at the host server.
The hacker will only have access to a heavily encrypted stream of data as it travels by.

To get the customers information a random hacker would require the following tools and abilities...

- Access to a router along the route from server to customer
- A sniffer and filter to listen to two way communications between a client and server
- The access point would have to positioned so that all packets during the conversation went by this point. (Internet packets by definition do not always travel the same route from point to point)

---

[1] Intermediate caching of pages at the client may be an inconvenience , the regeneration of the encryption key may be done hourly or daily . The frequency if dynamic key generation may be changed to fit the customers needs. If the keys are not re-generated with every transaction, the customer will experience a slightly higher risk of break in.

[2] Using a socket to send the credit number back to the server is not required. Any mechanism may be used to send the credit card number from the client browser to the server. The key point is that a Java applet does the encryption algorithm on the client browser.

    - Would have to identify the conversation from server to client as containing sensitive information from its context.
    - Would have to pick off the encrypted stream of sensitive data going back to the server.
    - Would have to be an expert in breaking encrypted data with private keys.
- The encrypted data would be complex enough that brute force guessing the key will require trillions of years using the most powerful computing resources not yet invented, so the hacker will have to eliminate some combinations.
- The hacker would only get one piece as sensitive information for his/her effort.

The moral of the story is that breaking directly into the server is much easier than cracking a transient partially encrypted conversation on the internet, and would produce more abundant information. The sparse encryption method does address fixed server security issues.

Q: I've heard that Java is slow and has bugs

A: Using bare minimum Java Apps the risk of hitting a browser bug are very minimal.

Q: What about using Netscape and Microsoft SSL browsers and servers?

They are the standard and if processing power at the server is not an issue they should be used.
Sparse technology is an optimization to reduce server costs for the same class of service as SSL.
I cannot argue the relative security of the SSL methodology compared with sparse security, I don't know if changing the keys dynamically with every transaction is employed by SSL??? Even if SSL technology is 5 times tougher to crack (no security is perfect) than the sparse encryption method, the weak link in internet transaction will continue be the server, and the trusted its human administrators, not the transient data on the internet itself.

EXHIBIT B

Penrod, Jack R (Jack)
From: Eric Grosse [SMTP:ehg@bell-labs.com]
Sent: Monday, September 29, 1997 9:45 AM
To: areisman@lucent.com
Subject: Re: security idea

The basic idea of setting up an encryption scheme with an on/off mode is a good one, and worth patenting if you can get away with it. I don't recall hearing it mentioned anywhere in print, though it does seem like a fairly obvious optimization.

The specific application of credit cards in Web forms and Java applets is perhaps not the best embodiment. SSL is already fine for such forms, and the main threat is card numbers being stolen off the remote server. Also, the sort of person paranoid about wiretapping is also reluctant to enable Java in general browsing.

Penrod, Jack R (Jack)
From: Art Reisman [SMTP:areisman@lucent.com]
Sent: Tuesday, September 30, 1997 9:52 AM
To: penrod@lucent.com
Subject: Security IDea for patent

Sparse Web Encryption with lowest common denominator Java

Art Reisman 614-855-9320
Lucent Technologies

Sparse-

You only encrypt those pieces of information that require it (Passwords, Credit Card numbers, Phone Numbers. SSL servers generally encrypt everything requiring much more horse power on the server and client side.

Lowest Common Denominator Java-

Client Encryption Uses the most basic Java Utilities and a very simple Java application. This eliminates common client browser differences with full blown Java Applications. The delays experienced with larger Java Applications are not encountered
Two key encryption insures that only the financial/merchant institution has the decryption key.

Basic scenario

1. Customer logs into web site to purchase an item.
2. The customer decides what they want , and follows the link to an HTML order form page.
3. On the purchase form embedded in the HTML page is a Java Applet that presents the field for the customers credit card number. The remainder of the page, and all its graphics are standard HTML, (hence the term sparse encryption)
4. The Java applet contains the encryption algorithm and unique encryption key created dynamically created each time the page is accessed.
5. The JAVA applet opens a socket back to merchants server and sends the encrypted credit card number back to the server.
6. The server holds the unique matching decryption key for this transaction. It also passed a session id (cookie) to the client so it match the correct decryption key.
7. The server securely decrypts the credit card number and fills the customer order.

FAQ

Q : What if a hacker picks off the order form with the encryption key.

Can the hacker pretend to be the customer.

A: In this scenario the hacker would be able to see everything except the credit card number. The hacker cannot get the credit number as it is only decrypted at the host server.

I've heard that Java is slow and has bugs

Using bare minimum Java Apps the risk of hitting a browser bug are very minimal.

Q: What about using Netscape and Microsoft secure browsers

This method is available now. The browser security wars are on going now and are sure to require a settling period. Lucent has this technology under patent pending and is the best simplest bullet proof solution there is. This simple method will port forward to any mechanism future browser employ. This method is rooted in bare minimum JAVA applet technology.

Penrod, Jack R (Jack)
From: Art Reisman [SMTP:areisman@lucent.com]
Sent: Tuesday, September 30, 1997 4:14 PM
To: Penrod, Jack areisman@lucent.com
Subject: Re: Security IDea for patent

Penrod, Jack wrote:
>
> Art,
> Doesn't sparse encryption only ensure that you are telling an
> interceptor that this part of the meassage is encrypted and from context
> of non-encrypted part telling the interceptor if it is a credit card
> number (note credit card numbers start with one of a handful of known
> fields, or telephone numbers (known area codes). This will help find
> the key or validate the crack of the message. With the power of today's

I plan on using transient (one time) private and public keys for each transaction, this is part of the beauty of this... here is a hacker scenario

1) Hacker has access to a public router and is picking IP addresses for some station along that route. And assembling them on his/her browser.

2) All packages going to that the station are going through this router.

3) The hacker sees some sort of financial application go to this station.

4) The hacker also sees and understands that there is some encryption algorithm embedded in this page

5) The hacker now re-tools (quickly) his sniffer and listens for data going back to this particlular server from which this financial application came.

6) The hacker gets the message coming back with an encrypted field embedded within it.

7) He suspects it is a credit card from the context of the surrounds.

8) The hacker creates a reverse decryption program that will try to break the private key, he also has to guess the algorithm.

9) The domain for the private key is 10 to the power of 20 (or something where the most powerful computer to be invented would require years to run all combinations.

10) Some how the hacker guesses the private key and decrypts the all relevent fields

11) The hacker then charges this credit card to the limit and leaves mo trail as to where the goods or cash are sent.

> computers isn't this just asking for hackers to crack it?
> I hope your answer is no, by the way, but I need the reasons.
> Jack Penrod

Your comments are valid, and you can make the argument that
this method is not as secure as encrytping everything. As it turns out
it very hard to pick up a transient conversation on the internet even
when no encryption is involved, most theft takes place at the server.
(the server is a goldmine where all the credit card numbers are).

SSL technology (encrypting everything) puts an unneccessary horse power
burden on the server CPU. For my business case I want to minimize
server load. If this means a less secure transaction than standard SSL
I'll have to let my customers vote.

I could encrypt additinal fields surrounding the credit card number to
make it more secure, and maybe change their delivery order etc.

Penrod, Jack

From: Art Reisman[SMTP:areisman@lucent.com]
Sent: Tuesday, September 30, 1997 4:13 PM
To: Penrod, Jack areisman@lucent.com
Subject: Re: Security IDea for patent

Penrod, Jack wrote:
>
> Art,
> Doesn't sparse encryption only ensure that you are telling an
> interceptor that this part of the meassage is encrypted and from context
> of non-encrypted part telling the interceptor if it is a credit card
> number (note credit card numbers start with one of a handful of known
> fields, or telephone numbers (known area codes). This will help find
> the key or validate the crack of the message. With the power of today's

I plan on using transient (one time) private and public keys for each transaction, this is part of the beauty of this... here is a hacker scenario

1) Hacker has access to a public router and is picking IP addresses for some station along that route. And assembling them on his/her browser.

2) All packages going to that the station are going through this router.

3) The hacker sees some sort of financial application go to this station.

4) The hacker also sees and understands that there is some encryption algorithm embedded in this page

5) The hacker now re-tools (quickly) his sniffer and listens for data going back to this particlular server from which this financial application came.

6) The hacker gets the message coming back with an encrypted field embedded within it.

7) He suspects it is a credit card from the context of the surrounds.

8) The hacker creates a reverse decryption program that will try to break the private key, he also has to guess the algorithm.

9) The domain for the private key is 10 to the power of 20 (or something where the most powerful computer to be invented would require years to run all combinations.

10) Some how the hacker guesses the private key and decrypts the all relevent fields

11) The hacker then charges this credit card to the limit and leaves mo trail as to where the goods or cash are sent.

> computers isn't this just asking for hackers to crack it?
> I hope your answer is no, by the way, but I need the reasons.
> Jack Penrod

Your comments are valid, and you can make the argument that

this method is not as secure as encrytping everything. As it turns out it very hard to pick up a transient conversation on the internet even when no encryption is involved, most theft takes place at the server. (the server is a goldmine where all the credit card numbers are).

SSL technology (encrypting everything) puts an unneccessary horse power burden on the server CPU. For my business case I want to minimize server load. If this means a less secure transaction than standard SSL I'll have to let my customers vote.

I could encrypt additinal fields surrounding the credit card number to make it more secure, and maybe change their delivery order etc.

Penrod, Jack

From: Art Reisman[SMTP:areisman@lucent.com]
Sent: Tuesday, September 30, 1997 9:52 AM
To: penrod@lucent.com
Subject: Security IDea for patent

Sparse Web Encryption with lowest common denominator Java

Art Reisman 614-855-9320
Lucent Technologies 9/25/97

Sparse-

You only encrypt those pieces of information that require it (Passwords, Credit Card numbers, Phone Numbers. SSL servers generally encrypt everything requiring much more horse power on the server and client side.

Lowest Common Denominator Java-

Client Encryption Uses the most basic Java Utilities and a very simple Java application. This eliminates common client browser differences with full blown Java Applications. The delays experienced with larger Java Applications are not encountered

Two key encryption insures that only the financial/merchant institution has the decryption key.

Basic scenario

1. Customer logs into web site to purchase an item.

2. The customer decides what they want , and follows the link to an HTML order form page.

3. On the purchase form embedded in the HTML page is a Java Applet that presents the field for the customers credit card number. The remainder of the page, and all its graphics are standard HTML, (hence the term sparse encryption)

4. The Java applet contains the encryption algorithm and unique encryption key created dynamically created each time the page is accessed.

5. The JAVA applet opens a socket back to merchants server and sends the encrypted credit card number back to the server.

6. The server holds the unique matching decryption key for this transaction. It also passed a session id (cookie) to the client so it match the correct decryption key.

7. The server securely decrypts the credit card number and fills the customer order.

FAQ

Q : What if a hacker picks off the order form with the encryption key. Can the hacker pretend to be the customer.

A: In this scenario the hacker would be able to see everything except the credit card number. The hacker cannot get the credit number as it is only decrypted at the host server.

I've heard that Java is slow and has bugs

Using bare minimum Java Apps the risk of hitting a browser bug are very minimal.

Q: What about using Netscape and Microsoft secure browsers

This method is available now. The browser security wars are on going now and are sure to require a settling period. Lucent has this technology under patent pending and is the best simplest bullet proof solution there is. This simple method will port forward to any mechanism future browser employ. This method is rooted in bare minimum JAVA applet technology.

Penrod, Jack R (Jack)

From: Penrod, Jack
Sent: Tuesday, September 30, 1997 9:57 AM
To: Art Reisman
Subject: RE: Security IDea for patent

Art,

Doesn't sparse encryption only ensure that you are telling an interceptor that this part of the meassage is encrypted and from context of non-encrypted part telling the interceptor if it is a credit card number (note credit card numbers start with one of a handful of known fields, or telephone numbers (known area codes). This will help find the key or validate the crack of the message. With the power of today's computers isn't this just asking for hackers to crack it?

I hope your answer is no, by the way, but I need the reasons.

Jack Penrod

EXHIBIT C

list each individual form of novelty; done below (verify)

Sparse Web Encryption with lowest common denominator Java

Art Reisman
Lucent Technologies

, updated 10/7/98

Sparse- plus coordination (e.g., cookie) at server and client

specify exemplary transmission e.g., packet(s), fields, sensitive data/attributes

claim language

You only encrypt those pieces of information that require it (Passwords, Credit Card numbers, Phone Numbers. SSL servers generally encrypt everything requiring much more horse power on the server and client side.

Secure Socket Layer

Lowest Common Denominator Java-

specify + contrast closest prior art

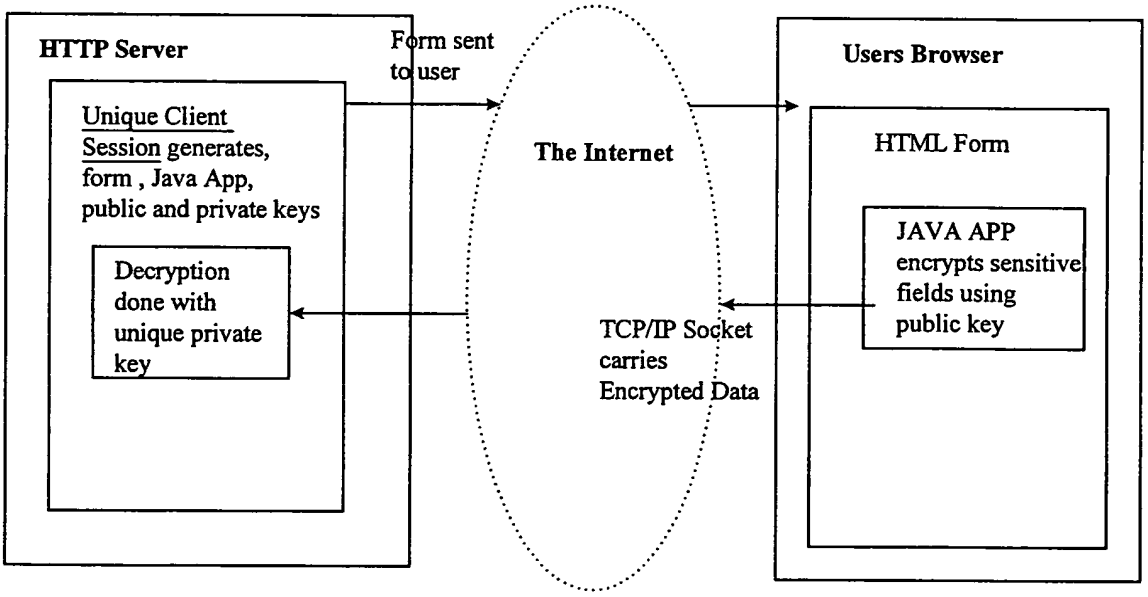
claim language

Client Encryption Uses the most basic Java Utilities and a very simple Java application. This eliminates common client browser differences with full blown Java Applications. The delays experienced with larger Java Applications are not encountered

Dynamic two key public/private encryption insures that only the financial/merchant institution has the decryption key.

; there a technical term as to this type of compatibility/ substantially universal browser compatibility

Sparse Encryption Scenario



Basic scenario

1. Customer logs into web site to purchase an item.

2. The customer decides what they want , and follows the link to an HTML order form page.

3. On the purchase form embedded in the HTML page is a Java Applet[1] that presents the sensitive data field(s) to the customer. The bulk of the page, and all its graphics are standard HTML, (hence the term sparse encryption)

4. The Java applet contains the encryption algorithm[2] and unique encryption public key created dynamically at the server each time the page is accessed. The server also dynamically creates a matching private key. The encryption algorithm will be chosen from the domain of available private/public key schemes and may not be unique to this implementation[3]

---

[1] Java is presently the predominant technology used for embedding computer programs in Web pages ; hence it is referenced here. Any machine independent web programming language supported on commercial Web browsers would suffice to perform the public key encryption on the client system, it need not be Java.

[2] Many public ~~public~~ Key, private key encryption schemes are available. For example the PGP algorithm covered under patent #4,405,829 could be adapted for the Java applet at the client for public key encryption; however it is more likely that Lucent will adapt one of their own algorithms for privaste and public key encryption.

Excerpt from Jeff Zimmerman
Boulder Software Engineering
3021 Eleventh Street
Boulder, Colorado 80304 USA
Phone: 303-541-0140

"In public key cryptosystems, everyone has two related complementary keys, a publicly revealed key and a secret key (also frequently called a private key). Each key unlocks the code that the other key makes. Knowing the public key does not help you deduce the corresponding secret key. The public key can be published and widely disseminated across a communications network. This protocol provides privacy without the need for the same kind of secure channels that a conventional cryptosystem requires.

Anyone can use a recipient's public key to encrypt a message to that person, and that recipient uses her own corresponding secret key to decrypt that message. No one but the recipient can decrypt it, because no one else has access to that secret key. Not even the person who encrypted the message can decrypt it."

[3] Intermediate caching of pages at the client may be an inconvenience , the regeneration of the encryption key may be done hourly or daily . The frequency of dynamic key generation may be changed to fit the customers needs. If the keys are not re-generated with every transaction, the customer will experience a slightly higher risk of break in.

EXHIBIT D

Lucent Technologies
Bell Labs Innovations

subject: Patent Submission 116621
(98-113) Sparse Web Encryption With Lowest Common Denominator Java

date: October 8, 1998
from: Jack R. Penrod
Org. P33A20000
IH 2A-419
(630) 979-2155
Corporate Counsel
Intellectual Property-Law

A. Reisman:

Your above-identified submission has been received and opened for further study. Please allow 2-4 weeks for me to review the subject matter and to contact you for further discussion.

If you know of any past or impending public disclosure of the subject matter (e.g., product release, publication in a journal or during a customer sales meeting), please let me know immediately.

All questions or comments in connection with this submission may be directed to me at 630-979-2155.

J. R. Penrod

IL0015-P33A20000-JRP-jea **J. R. Penrod**

Copy to
C. L. Warren
J. Zelasko

EXHIBIT E

SUBMISSIONS/CASES REFERRED TO OUTSIDE COUNSEL

Legal Secretary Completes: **IH-PAT** 98-113

IDS No. 116621

Case Name/No. _____

Classification Code IV
(Must Conform To Lucent Filing Policy)

Center P33A20000

Attorney J. R. Penrod

Filing Deadline _____

Date 2/22/99

Office Manager Completes:

Date Received From Legal Secretary 2/22/99

Date Sent To OC Staff _____

Subject Matter: Java Encryption

BU: OCTEL

Outside Counsel Staff Responsibilities: *Transferred in Alf:*

Step 1

Transfer to Outside Counsel _____ _____

Date Sent To Outside Counsel _____ _____

OR

Date Returned to Originating Center _____

Step 2 (Date received from Outside Counsel)

Upon receipt of Outside Counsel's recommendation, Outside Counsel Staff will either:

1. If defensive publication, follow Defensive Publication procedures _____
2. If application is being prepared, Outside Counsel Staff will enter the date they generated clsub and mkapp _____

PT380.oc (Rev. 4/97)
J:\forms\oc_transfer.doc 11/17/98